Procesamiento del Lenguaje Natural, num. 29 (2002), pp.213-220 recibido 28-05-2002; aceptado 02-07-2002

Modelado de diálogo basado en conocimiento, acciones y expectativas

José F. Quesada
Universidad de Sevilla
jquesada@cica.es

Resumen: Este trabajo propone un modelo declarativo para el modelado de sistemas de gestión de diálogo. La estrategia propuesta se basa en una división funcional del sistema de gestión en cuatro niveles: especificación, interfaz de usuario, representación y ejecución; cada uno de los cuales se corresponde, respectivamente con las estructuras regla de diálogo, movimiento de diálogo, estado de información y fase de diálogo. El trabajo se centra específicamente en el nivel de especificación y describe la capacidad de dirección por conocimiento, el control de expectativas y la ejecución de acciones.
Palabras clave: Sistemas de Diálogo Hablado.

Abstract: This work proposes a declarative model for the modelling of dialogue management systems. The model is based on a functional division which distinguishes four main levels: specification, user-interface, representation and execution, which correspond, respectively, to the structures dialogue rule, dialogue move, information state and dialogue phase. The work concentrates on the specification level and describes its main features: knowledge-driven, control of expectations and execution of actions.
Keywords: Spoken Dialogue Systems.

1 Introducción

Un análisis exhaustivo del diálogo requiere tener en cuenta tanto los aspectos relacionados con la representación semántica del contenido de las intervenciones de los diferentes participantes como la información pragmática asociada (Scott y Kamp 1995). Esta información pragmática incluye los actos de habla, la información disponible para cada participante, los objetivos de la propia proferencia, etc.

El diseño de sistemas computacionales para la gestión del diálogo o interfaces computacionales (Zue y Glass 2000) debe por tanto cubrir ambos aspectos. Es decir, debe incorporar tanto un formalismo para la representación semántica del lenguaje natural como un modelo para la manipulación de la información pragmática. Asimismo, este último componente debe permitir una gestión adecuada del carácter dinámico del propio discurso (Kamp y Reyle 1993, Barwise y Perry 1983).

Dentro del ámbito de la Lingüística Computacional y la Ingeniería del Lenguaje, una línea de investigación relacionada con el área del diálogo se ha centrado en el modelado del diálogo, asumiendo entre otros objetivos el desarrollo de algoritmos, estrategias y formalismos que permitan la participación de un sistema computacional en un diálogo cooperativo.

Durante los últimos años se han desarrollado varias teorías y estrategias para el modelado del diálogo, entre las que cabe citar:

- Las *gramáticas de diálogo*, basadas en la aplicación de técnicas de estados finitos a la gestión del diálogo. Los sistemas *Speech Acts* (Martin *et al* 1996) de Sun y *Train Timetable* (Aust *et al* 1995) de Philips utilizan este modelo). *CSLU rapid prototyper* (Sutton et al 1996) utiliza un enfoque similar basado en redes de transición de estados.
- Modelos de diálogo basados en *planes* (Carberry 1990, Sadek 1991), donde los diálogos se analizan como instancias de modelos (planes) definidos en un meta-

ISSN 1135-5948

nivel. El sistema *Atlas* (Freeman 2000) utiliza un modelo de planes, mientras que la Teoría de Juegos Conversacionales (Kowkto *et al* 1991) está basada en planes e incorpora gramáticas de diálogo. Los sistemas basados en agenda, tales como el *CMU Communicator* (Xu y Rudnicky 2000) se pueden incluir en esta categoría, en tanto que la agenda se entiende como una especificación del plan global necesario para llevar a cabo una tarea.

- Enfoques basados en un *modelo de acción conjunta* (Grosz y Kraus 1993, Lochbaum 1994) que asume una teoría de la conversación como actividad conjunta resultado de la colaboración entre agentes. Relacionado con este grupo se encuentran los enfoques colaborativos, entre los que cabe citar *TRAINS* (Ferguson et al 1996).
- Enfoques basados en la *actualización del estado de información* (Traum *et al* 1999), donde se contempla el diálogo como un proceso de actualización dinámica de una estructura semántica (el estado de información).

Este trabajo describe un formalismo para el modelado de sistemas de gestión de diálogo. El modelo propuesto pretende cubrir los siguientes objetivos:

- A un nivel formal, cubre la motivación que se encuentra tras la teoría de la acción conjunta y el enfoque basado en la actualización del estado de información. En concreto, el análisis propuesto para fenómenos cooperativos tales como subdiálogos de ayuda, confirmaciones, tratamiento flexible de patrones de parámetros, etc., se encuentra en la línea de la teoría de la acción conjunta. Por otro lado, el modelo implementa directamente la noción de estado de información, y tanto la historia del diálogo como la propia gestión, dirección y colaboración en el diálogo están controladas por los estados de información.
- Utiliza un enfoque declarativo para la especificación del modelo de diálogo. Este enfoque pretende superar las limitaciones de los sistemas basados en gramáticas de diálogo y redes de transición de estados (flexibilidad, naturalidad, capacidad explicativa, complejidad en la especificación de mecanismos de reparo, etc.) y en planes (complejidad inferencial, flexibilidad, etc.).
- El proceso de diálogo se contempla como un fenómeno cooperativo dirigido tanto por la representación semántica de la información suministrada por el usuario como por las expectativas de diálogo generadas por el propio sistema de gestión.

La sección 2 introduce los principales componentes del modelo de diálogo, específicamente los 4 niveles básicos y su correlación con una arquitectura genérica para el diseño e implementación de sistemas de gestión de diálogo hablado.

A continuación, la sección 3 describe el nivel de especificación del modelo de diálogo basado en la noción de regla de diálogo, la cual incluye como componentes básicos las condiciones de disparo basadas en conocimiento (*TriggeringConditions*), las pre- y post-acciones (*PreActions*, *PostActions*), el control de las expectativas (*DeclareExpectations* y *ActionsExpectations*) y la gestión del estado de información.

2 Principales componentes para el modelado y gestión del diálogo

Para lograr una mayor modularidad y facilitar la implementación de los sistemas de gestión de diálogo, el modelo de diálogo se divide en 4 componentes o niveles básicos: especificación, interfaz de usuario, representación y ejecución (Quesada *et al* 1991).

2.1 Nivel de especificación

Este módulo implementa en realidad un lenguaje de especificación para el diseño y modelado de sistemas de diálogo. Está orientado hacia la especificación a alto nivel de las características y funciones del sistema. La noción o estructura básica es la de regla de diálogo (***Dialogue Rule***). Este nivel será descrito en la sección 3.

2.2 Nivel de interfaz de usuario

Este nivel se encarga de gestionar las interrelaciones entre el sistema de gestión de diálogo y el usuario. En sistemas avanzados este nivel permite incorporar funciones de mejora de los sistemas de reconocimiento y síntesis de voz incluyendo la información semántica y pragmática del modelo de diálogo. Asimismo, este nivel se encargará de funciones tales como la consolidación y organización de los movimientos de diálogo: repeticiones, orden

de las intervenciones, subdiálogos, interrupciones críticas, etc.

Para su funcionamiento, este nivel incorpora una estructura intermedia entre el módulo de entrada o reconocimiento de voz y el gestor de diálogo. Esta estructura (*InputPool*) será manipulada por el módulo de selección de movimientos de diálogo (*Dialogue Move Selector*) para reorganizar los movimientos de diálogo pendientes, eliminar repeticiones, modificar las precedencias debido a criticidad o presencia de subdiálogos, etc.

La noción o estructura básica del nivel de interfaz de usuario es la de movimiento de diálogo (***Dialogue Move***): toda proferencia de usuario será analizada y representada semánticamente como uno o más movimientos de diálogo, y toda respuesta del gestor de diálogo es representada como uno o más movimientos de diálogo, los cuales serán procesados por los módulos de generación y síntesis (pertenecientes asimismo a este nivel de interfaz de usuario).

2.3 Nivel de representación

El principal cometido de este nivel es la manipulación y gestión de la historia del diálogo, permitiendo resolver fenómenos propios de la estructura del diálogo mismo como son las referencias anafóricas, o bien de la gestión del diálogo como son las expectativas.

La noción o estructura básica de este nivel es la de estado de información (***Information State***). Los distintos estados de información que constituyen la historia se almacenan en un repositorio al que pueden acceder el resto de módulos del sistema.

2.4 Nivel de ejecución

Este nivel está formado por el gestor de diálogo. Este módulo recibe por un lado el modelo de diálogo (reglas de diálogo) del nivel de especificación y por otro los movimientos de diálogo provenientes del nivel de interfaz. Con esta información y la disponible en el nivel de representación (estados de información de la historia del diálogo) se encarga de realizar las operaciones correspondientes: generar salidas a través del nivel de interfaz, resolver anáforas o expectativas usando la historia previa del diálogo, incorporar nuevos estados en la historia, ejecutar las acciones pertinentes en función de los estados de información resueltos, etc.

Este nivel puede incorporar módulos especializados en la gestión del conocimiento que permitan por ejemplo la resolución de construcciones referenciales en dominios específicos y en gestión de acciones, encargados asimismo de la ejecución de acciones propias del dominio de aplicación del sistema de gestión de diálogo.

La noción o estructura básica del nivel de ejecución es la de fase de diálogo (***Dialogue Phase***). Una fase de diálogo (*Dphase*) es la instaciación de una regla de diálogo (*Drule*) para un movimiento de diálogo (*Dmove*), creando consecuentemente un estado de información (*Istate*):

Dphase = Drule + Dmove + Istate

3 Modelado de diálogo basado en conocimiento, expectativas y acciones

La especificación de un sistema de diálogo está basada en un conjunto de estructuras, denominadas reglas de diálogo, cada una de las cuales es independiente del resto de estructuras (principio de la especificación declarativa).

3.1 Modelo dirigido por conocimiento

Uno de los principales mecanismos de activación de las reglas es la información (representación semántica) que llega al módulo de gestión de diálogo (nivel de ejecución) desde el interfaz de usuario. El formalismo para la representación está basado en estructuras complejas de rasgos y la selección de reglas de diálogo se hace mediante unificación con el campo *TriggeringConditions* de las reglas de diálogo.

En caso de que exista más de una regla de diálogo que cumple con las condiciones de unificación se utilizará el rasgo *PriorityLevel* para indicar el orden de precedencia de las reglas (a mayor valor numérico menor nivel de prioridad).

Utilizando el modelo de representación semántica para sistemas de gestión de diálogo hablado basado en lenguajes de instrucciones naturales (Amores y Quesada 2000), la sentencia

In[1]: "Querría hacer una llamada telefónica."
se representaría mediante:

```
DMOVE: specifyCommand
TYPE:  TelephoneCall
ARGS:  [Destination]
CONTENT:-
```

Suponiendo que la especificación del sistema de diálogo contiene, entre otras, las siguientes reglas:

```
(RuleID: TELEPHONECALL,
 TriggeringConditions:
    (DMOVE:specifyCommand,
     TYPE :TelephoneCall),
 PriorityLevel: 10,
 ...)

(RuleID: REDIAL,
 TriggeringConditions:
    (DMOVE:specifyCommand,
     TYPE: Redial),
 PriorityLevel: 10,
 ...)

(RuleID: SAFETYNET,
 TriggeringConditions:
    (),
 PriorityLevel: 100,
 ...)
```

podemos observar que tanto las reglas *TELEPHONECALL* como *SAFETYNET* se activarían por unificación con la estructura de rasgos anterior y las correspondientes *TriggeringConditions* de cada regla. Ahora bien, el nivel de prioridad de *TELEPHONE-CALL* es superior al de *SAFETYNET* (esta regla está ideada como un mecanismo de robustez del gestor de diálogo que capture cualquier entrada que sea rechazada por el resto de reglas).

La activación de una regla de diálogo (*Drule*) crea una instancia de dicha regla en el nivel de ejecución denominada fase de diálogo (*Dphase*) que asocia con la regla el estado de información (*Istate*) que contendrá inicialmente el movimiento de diálogo (*Dmove*) utilizado para la activación de la regla.

3.2 Control y dirección por expectativas

Si la iniciativa del usuario es capturada y manipulada por el modelo propuesto a través de la capacidad de dirección por conocimiento explicada en el apartado anterior, la iniciativa del sistema se modela mediante el uso de expectativas.

El rasgo *ARGS* especifica el conjunto de atributos que un estado de información debe contener para considerarse coherente.

Formalmente, se utiliza un modelo tipo disyunción de conjunciones para especificar múltiples modelos de atributos subcate-gorizados. El principio de coherencia indica que para cada valor de ARGS debe existir un rasgo instanciado con dicho nombre en la propia estructura. Este principio de coherencia se aplica recursivamente, permitiendo que la coherencia de una estructura dependa composicionalmente de la coherencia de sus partes.

Cada uno de los rasgos que generan incoherencia en un estado de información originan una expectativa con el mismo nombre que el propio rasgo en la fase de diálogo correspondiente al estado de información.

Si consideramos el estado de información obtenido en el ejemplo anterior podemos observar que el rasgo *ARGS* incluye el parámetro *Destination* y el estado de información no incluye dicho rasgo.

Para cada regla de diálogo, el campo *ActionsExpectations* permite indicar qué acciones se deben ejecutar en caso de que se activen determinadas combinaciones de expectativas.

Por ejemplo, la regla *TELEPHONECALL*:

```
(RuleID: TELEPHONECALL,
 TriggeringConditions:
    (DMOVE:specifyCommand,
     TYPE :TelephoneCall),
 PriorityLevel: 10,
 ActionsExpectations: {
    [Destination]:
      UserPrompt("¿A quien
                  quiere llamar?");
 }
 DeclareExpectations: {
    Destination <= DESTINATION;
 }
 ...)
```

indica que se le pregunte al usuario el destino de la llamada (la función *UserPrompt* permite enviar al interfaz de salida sentencias ya generadas para que el módulo de síntesis las reproduzca directamente).

Por otra parte, el campo *Declare-Expectations* indica cómo resolver una expectativa. Existen dos mecanismos para la resolución.

El primero permite especificar un modelo de rasgos que un movimiento de diálogo debe

cumplir para completar el rasgo que origina la expectativa. Este modelo se puede considerar como una aplicación de la resolución de condiciones de disparo de las reglas a las expectativas.

El segundo mecanismo permite especificar una regla de diálogo como condición de resolución de la expectativa. De esta forma, si en un estado posterior del diálogo se consigue generar dicha regla (y ésta pasa el control de coherencia), el estado de información de la fase correspondiente se fusionará con el estado de información de la fase que provocó la expectativa originando una nueva fase de diálogo asociado a la misma regla original pero con un estado de información actualizado con la nueva información.

Continuando con el ejemplo anterior, la regla *TELEPHONECALL* especifica en el campo *DeclareExpectations* que el rasgo *Destination* se resuelva con la regla *DESTINATION*. Supondremos que a la pregunta generada por el sistema:

Out[1]: "¿A quien quiere llamar?"

el usuario responde:

In[2] "Al número 123456789."

que generará el siguiente movimiento de diálogo:

```
DMOVE:  specifyParameter
TYPE:   Destination
ARGS:   []
CONTENT:123456789
```

Este movimiento de diálogo es capturado por la regla *DESTINATION*:

```
(RuleID: DESTINATION,
 TriggeringConditions:
    (DMOVE: specifyParameter,
     TYPE:  Destination),
 PriorityLevel: 5)
```

Puesto que esta estructura es coherente (el campo *ARGS* está vacío), la fase se considera completa, y el estado de información de esta fase pasa a formar parte del rasgo *Destination* del estado de información de la fase anterior, creando una nueva fase de diálogo asociada a la misma regla pero con un nuevo estado de información:

```
DMOVE:  specifyCommand
TYPE:   TelephoneCall
ARGS:   [Destination]
CONTENT:-
Destination :
    DMOVE:  specifyParameter
    TYPE:   Destination
    ARGS:   []
    CONTENT:123456789
```

3.3 Control de la ejecución de acciones

Gran parte de la utilidad de los sistemas de gestión de diálogo se basa en la capacidad que éstos poseen para ejecutar acciones. En el sistema que estamos presentando, se considera acción a toda función que el sistema de gestión de diálogo realiza, bien directamente o a través de módulos, recursos o agentes externos.

El sistema incluye un conjunto básico de acciones, tanto de tipo interno como externo al propio gestor de diálogo.

3.3.1 Acciones internas al gestor de diálogo

- ***UserPromp***: genera una salida al usuario enviándola directamente al módulo de síntesis de voz.
- ***UserGeneration***: el objetivo es generar una salida al usuario pero a través de un módulo intermedio de generación de lenguaje natural que utiliza como entrada un estado de información.
- ***ActivateRule***: permite activar una regla de diálogo directamente. Este mecanismo de activación de reglas de diálogo es un mecanismo fuerte de enlace entre reglas de diálogo (distinto al modelo de activación dirigido por conocimiento: representación semántica de las proferencias del usuario). Mediante este mecanismo se pueden modelar fenómenos tipo gramáticas de diálogo o estrategias de redes de estados finitos.
- ***CreateInformationState***: permite crear un estado de información asociado a la fase de diálogo en la que se ha activado la regla correspondiente. Este mecanismo complementa la funcionalidad de *Activate-Rule* y mediante su uso es posible resolver fenómenos tales como *Task Accomodation*

(creando un estado de información inicial que actuaría como modelo por defecto).

- ***AnaphorResolution***: si la representación semántica incluye marcas anafóricas, el gestor de diálogo puede llevar a cabo la resolución utilizando una estrategia basada en unificación. La resolución se realiza sobre los estados de información asociados a las fases de diálogo anteriores a la fase actual.

3.3.2 Acciones externas al gestor de diálogo

- ***ReferenceResolution***: para cada dominio aparecerán parámetros específicos cuya resolución deberá realizarse mediante un gestor de conocimiento (*Knowledge Manager*) especializado en el dominio. Por ejemplo, determinar a qué número de teléfono se corresponde la sentencia "*Ponme con la secretaria de Juan Rodríguez*" requerirá acceder a un sistema de directorio donde se pueda resolver la referencia correspondiente a la expresión "*la secretaria de Juan Rodríguez*".
- ***ExecuteAction***: En este caso, el sistema de diálogo podrá conectarse con un módulo específico de control de acciones (*Action Manager*) que ejecute la operación asociada a la representación semántica del estado de información correspondiente. Al igual que ocurre con el *Knowledge Manager*, el *Action Manager* es un módulo propio del dominio de aplicación del sistema de gestión de diálogo.

3.3.3 Pre- y post-acciones

El modelo de especificación permite dos momentos de ejecución de acciones:

- Las pre-acciones (*PreActions*) se ejecutan al activarse la regla (bien por disparo de las *TriggeringConditions* o bien por activación mediante *ActivateRule*). Las pre-acciones se ejecutan por tanto antes de iniciarse la parte correspondiente a análisis de coherencia y control de expectativas.
- Las post-acciones (*PostActions*) se ejecutan una vez que se ha superado la fase de análisis de coherencia de la regla, es decir, una vez superadas todas las expectativas de la regla.

Continuando con la regla *TELEPHONE-CALL*, en la siguiente especificación podemos observar cómo la resolución anafórica se ha situado en las pre-acciones. Si hay referencias anafóricas será necesario resolverlas antes de analizar las expectativas ya que la propia resolución anafórica puede resolverlas. Por ejemplo, la frase "*Llámalo*" contiene una referencia anafórica, pero no un destino de la llamada. De esta forma, antes de provocar que el sistema pregunte a quién desea llamar, conviene resolver la anáfora, ya que este proceso incluirá el destino de la llamada. En las post-acciones se ha colocado la resolución de referentes (que determine el número de teléfono al que se hará la llamada en caso de que llegue no directamente un número sino una descripción de dicho número), y a continuación, la ejecución de la acción:

```
(RuleID: TELEPHONECALL,
 TriggeringConditions:
    (DMOVE:specifyCommand,
     TYPE :TelephoneCall),
 PriorityLevel: 10,
 PreActions: {
    AnaphorResolution();
 }
 ActionsExpectations: {
    [Destination]:
      UserPrompt("¿A quien
                  quiere llamar?");
 }
 DeclareExpectations: {
    Destination <= DESTINATION;
 }
 PostActions : {
    ReferenceResolution();
    ExecuteAction(call,
           @is.Destination.CONTENT);
 } )
```

4 Conclusión y trabajo futuro

Este trabajo ha presentado una estrategia para el modelado de sistemas de gestión de diálogo basado en el principio de especificación declarativa, que intenta superar las limitaciones de los modelos basados en gramáticas de diálogo, estados finitos y planes, incorporando las principales ventajas de los sistemas basados en los modelos de acción conjunta y el enfoque de la actualización del estado de información.

Esta estrategia de modelado está íntimamente integrada con una arquitectura de diseño basada en cuatro niveles funcionales: especificación, interfaz de usuario, representación y ejecución.

La estrategia en sí permite la dirección por conocimiento del sistema de gestión, la manipulación y gestión de expectativas y la ejecución de acciones.

Como conclusión, merece la pena destacar cómo el modelo propuesto integra tanto un formalismo para la representación semántica como una estrategia para la manipulación pragmática (Scott y Kamp 1995, Zue y Glass 2000, Kamp y Reyle 1993, Barwise y Perry 1983).

El modelo ha sido utilizado para el diseño de varios sistemas de gestión de diálogo para lenguajes naturales de instrucciones, en concreto, para un sistema de diálogo hablado en el dominio de la domótica y otro en el dominio de la gestión de sistemas telefónicos automáticos, desarrollados en los proyectos Dhomme (IST-2000-26280) (Quesada *et al* 2001) y Siridus (IST-1999-10516) (Amores y Quesada 2000).

Entre las principales líneas de trabajo futuro merece la pena destacar la incorporación de nuevos modelos de diálogo, tales como centrados en tarea (*task-centred*) o negociativos.

Bibliografía

Amores, J. G., Quesada, J. F. 2000. *Dialogue Moves in Natural Command Languages*. Siridus project. Deliverable 1.1. http://www.ling.gu.se/projekt/siridus/

Aust, H., Oerder, M., Seide, F., Steinbiss, V. 1995. The Philips automatic train timetable information system. *Speech Communication* **17**, 249-262.

Barwise, J., J. Perry. 1983. *Situations and Attitudes*, MIT Press, Cambridge, MA.

Carberry, S. 1990. *Plan recognition in natural language dialogue.* ACL-MIT Press Series in Natural Language Processing. Bradford Books, MIT Press.

Ferguson, G. M., Allen, J. F., Miller, B. W., Ringger, E. K. 1996. The Design and Implementation of the TRAINS-96 System: A Prototype Mixed-Initiative Planning Assistant. *TN96-5*, Computer Science Dept., U. Rochester.

Freedman, R. 2000. A Reactive Approach to Dialogue Planning in an Intelligent User Interface. *Workshop on Using Plans in Intelligent User Interfaces, International Conference on Intelligent User Interfaces (IUI 2000)*, New Orleans.

Giachin, E., McGlashan, S. 1997. Spoken Language Dialogue Systems. En Young, S., Bloothooft (eds.) *Corpus-based methods in language and speech processing*. Dordrech: Kluwer Academic Publishers, pp. 69-117.

Gibbon, D., Moore, R., Winski, R. (eds.) 1997. *Handbook of Standards and Resources for Spoken Language Systems*. New York: Mouton de Gruyter.

Grosz, B., Kraus, S. 1993. Collaborative plans for group activities. *Proceedings of IJCAI-93*, volumen 1, pages 367-373.

Kamp, H., Reyle, U. 1993. *From Discourse to Logic*. Kluwer, Dordrecht.

Kowtko, J. , Isard, S. , Doherty, G. M. 1991. Conversational games within dialogue. En M. Caenepeel, J. L. Delin, L. Oversteegen, G. Redeker y J. Sanders, eds. *Proceedings of the DANDI Workshop on Discourse Coherence.*

Lochbaum, K. E. 1994. *Using Collaborative Plans to Model the Intentional Structure of Discourse.* PhD thesis, Harvard University.

Luperfoy, S. (ed.) 1998. *Automated Spoken Dialog Systems.* Cambridge: MA: MIT Press.

Maier, E., Mast, M., Luperfoy, S. (eds) 1997. *Dialogue Processing in Spoken Language Systems*. Springer-Verlag.

Martin, P., Crabbe, F., Adams, S., Baatz, E., Yankelovich, N. 1996. SpeechActs: A Spoken Language Framework. *IEEE Computer*, **29(7)**.

Quesada, J. F., Amores, J. G., Bos, J., Ericsson, S., Gorrell, G., Knight, S., Lewin, I., Milward, D., Rayner, M. 2001. *Configuring Linguistic Components in a Plug and Play Environment.* Dhomme project. Deliverable 3.1.(http://www.ling.gu.se/projekt/dhomme/)

Rickel, J., Ganeshan, R., Rich, C., Sidner, C.L., Lesh, N. 2000. Task-Oriented Tutorial Dialogue: Issues and Agents. *AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications.* Technical Report FS-00-01, November 2000, pp 52-57.

Sadek, D. 1991. Dialogue acts are rational plans. *Proceedings of the ESCA/ETRW Workshop on the Structure of Multimodal Dialogue*, Maratea.

Scott, D., Kamp, H. 1995. Dialogue Modelling. En Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A., Zue, V., Varile, G., Zampolli, A. (eds.). Survey of the State of the Art in Human Language Technology. http://cslu.cse.ogi.edu/HLTsurvey/

Sutton S., Vermeulen, P., de Villiers, J., Schalkwyk, J., Fanty, M., Novick, D., Cole, R. 1996. *Technical Specification of the CSLU Toolkit.* Technical Report No. CSLU-013-96, Center For Spoken Language Understanding, Oregon Graduate Institute of Science & Technology.

Traum, D., J. Bos, R. Cooper, S. Larsson, I.Lewin, C. Matheson and M. Poesio. 1999. *A model of dialogue moves and information state revision.* Deliverable 2.1. Trindi project http://www.ling.gu.se/projekt/trindi/

Xu, W., Rudnicky, A. 2000. Task-based dialog management using an agenda. *ANLP / NAACL 2000 Workshop on Conversational Systems*, pp. 42-47.

Zue, V., Glass, J. 2000. Conversational Interfaces: Advances and Challenges. Invited Paper, *Special issue on Spoken Language Processing, Proc. IEEE*, **88(8),** pp. 1166-1180.